Biol Cybern (2015) 109:561–574
DOI 10.1007/s00422-015-0661-7

ORIGINAL ARTICLE

Biological Cybernetics

Cerebellum-inspired neural network solution of the inverse kinematics problem

Mitra Asadi-Eydivand[1,2] · Mohammad Mehdi Ebadzadeh[2] · Mehran Solati-Hashjin[3] · Christian Darlot[4] · Noor Azuan Abu Osman[1]

Received: 4 June 2014 / Accepted: 23 September 2015 / Published online: 5 October 2015

Abstract The demand today for more complex robots that have manipulators with higher degrees of freedom is increasing because of technological advances. Obtaining the precise movement for a desired trajectory or a sequence of arm and positions requires the computation of the inverse kinematic (IK) function, which is a major problem in robotics. The solution of the IK problem leads robots to the precise position and orientation of their end-effector. We developed a bioinspired solution comparable with the cerebellar anatomy and function to solve the said problem. The proposed model is stable under all conditions merely by parameter determination, in contrast to recursive model-based solutions, which remain stable only under certain conditions. We modified the proposed model for the simple two-segmented arm to prove the feasibility of the model under a basic condition. A fuzzy neural network through its learning method was used to compute the parameters of the system. Simulation results show the practical feasibility and efficiency of the proposed model in robotics. The main advantage of the proposed model is its generalizability and potential use in any robot.

✉ Mitra Asadi-Eydivand
mitra@um.edu.my

1 Department of Biomedical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia

2 Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran 15914, Iran

3 Department of Biomedical Engineering, Amirkabir University of Technology, Tehran 15914, Iran

4 Département de Traitement des signaux et des images, Ecole Nationale Supérieure des Télécommunications, 75634 Paris Cedex 13, France

Keywords Inverse kinematics · Cerebellar neural network · Cerebellar cortex · Robot manipulator · Bioinspired model · Fuzzy neural network

1 Introduction

Robots are widely used primarily in industrial and medical applications where responsible, stable, and highly accurate operations are required. The demand today for more complex robots that have manipulators with higher degrees of freedom (Dof) is increasing because of technological advances. Obtaining the precise movement for a desired trajectory or sequence of arm and positions requires the computation of the inverse kinematic (IK) function, which is a major problem in robotics (Alavandar and Nigam 2008; Köker 2013; Wu and Rao 2007).

Forward kinematics involves determining the position of the end-effector of the robot given its joint variables. Obtaining the joint variable of a robot manipulator, given the desired position of the end-effector of the robot, is called IK (de Jesús Rubio et al. 2013; Zhang and Paul 1991). The solution of the IK problem requires the real-time computation and uniqueness of the inverse function. The solution of the IK problem has been studied by many researchers (Ali et al. 2010; Kanoun et al. 2011; Kumar et al. 2010; Reinhart and Steil 2011; Wang et al. 2010). Many approaches to solving the IK problem can be categorized into (1) analytical-based and (2) learning-based methods.

Although both types of methods can efficiently solve the problem, they have several shortcomings. First, the computations of the complex models in both methods are time-consuming because of the complexity of the mathematical formulation. Furthermore, poor efficiency results from the failure of the model to specify the robot characterization.

By contrast, if the model can reflect all of the robot characterization, then the results are more specific. Second, the learning data generated from the IK function in a restricted domain in most learning-based methods result in limited convergence. Neither method proposes a general solution for IK problems (Hasan et al. 2006, 2010).

The proposed solution can be categorized into a learning-based method in the present study. However, two major distinctions can be recognized. First, this study proposes a new artificial neural network (ANN) model inspired by the anatomy of the cerebellum. Second, the IK function is learned using the forward kinematics function, unlike in other studies.

Different solutions to the IK problems of serial manipulators through learning-based methods exist. Among these solutions, learning-based methods that use ANNs are similar to the method used in the present study.

The approach used by Alavandar and Nigam (2008) in solving the IK problem is based on an adaptive neuro-fuzzy interface system, which was used to predict and estimate the problem. The said researchers collected training data from the forward kinematics of the 2-Dof and 3-Dof robot manipulator to show the effectiveness of this approach. The obtained results are inconsistent with the analytical IK function. The disadvantage of this method is its limited convergence because collecting data from forward kinematics covers only a part of the IK function domain.

Karlik and Aydin (2000) implemented a robot manipulator with six Dof through the best ANN configuration from two ANNs: a three-layer back-propagation (BP) with six outputs and six four-layer BPs with one output. The error obtained from the second ANN is smaller than that from the first configuration. The data set elements (i.e., inputs and outputs) for training NNs were calculated from analytic equations based on restrictions.

Xia and Wang (2001) developed a recurrent NN called the dual network, which has a single neuron layer. The approach proposed by Arefi and Sadigh (2011) is based on a fuzzy algorithm and a polar coordinate displacement of the robot manipulator's tip. The model overcomes blind spots and singularities. NNs based on the radial basis function and multilayer perceptron, used to predict incremental joint angles, were proposed by Chiddarwar et al. and K.K. Dash et al., respectively (Chiddarwar and Ramesh Babu 2010; Dash et al. 2011). A genetic algorithm was used to improve the accuracy of the ANN model in several of these studies (Köker 2013; Oyama et al. 2001).

The ability of biological motor control systems to fine-tune themselves and enhance their performance is one of their major inspiring aspects. The cerebellum is indispensable in achieving fast and precise coordinated movements and accurately perceiving body motion; thus, the use of the cerebellum as a model for control and learning movement has attracted the attention of researchers for many years (Albus 1975b; Darlot 1993; Eccles et al. 1967; Kawato et al. 1987; Miall et al. 1993).

The current study attempts to modify the shortcomings of previous models of cerebellar NNs, which is more consistent with the general anatomy and functioning of the cerebellar pathways to solve the IK problem. Moreover, the disadvantage of previous cerebellar NN models and strong point of the proposed model have been mathematically proven.

2 Method

2.1 Cerebellar NN Model

Computing an inverse function is an "ill-posed problem" with no general solution, except in trivial cases (Cannon and Robinson 1987; Tikhonov and Arsenin 1979).

The forward kinematics function is shown in Eq. (1), where $\boldsymbol{\theta}(\mathbf{t}) = (\theta_1(t), \theta_2(t), \ldots \theta_n(t))$ is the joint variable of a manipulator with n Dof at any instant of time. The position variable at any instant of time in the x, y, and z directions is denoted by $\mathbf{P}(\mathbf{t}) = (x(t), y(t), z(t))$, and f is a nonlinear function.

$$\mathbf{P}(\mathbf{t}) = f(\boldsymbol{\theta}(\mathbf{t})) \tag{1}$$

By contrast, the IK function can be computed using Eq. (2). However, the inverse function (f^{-1}) is not unique, and the set of solution is infinite because of the nonlinear, uncertain, and time-varying nature of f.

$$\boldsymbol{\theta}(\mathbf{t}) = f^{-1}(\mathbf{P}(\mathbf{t})) \tag{2}$$

A direct function is deterministic, whereas an inverse function is not necessarily so. Therefore, similar effects can be induced by different sets of causes. For instance, the biomechanical function of the arm, which expresses movement caused by exerting forces, is deterministic according to Newton's law. Conversely, the same hand displacement can result from various configurations of articulated arm segments and can be induced at different levels of stiffness of articulations. Given that different causes can produce similar effects, a cause–effect relationship is generally not bijective but is rather a surjection from the domain of the causes to that of the effects. Therefore, no general method permits a definite return from an effect to a single cause, and finding an inverse function is a process that is very sensitive to the initial conditions and noise. Thus, in practice, an inverse function that is appropriate at one instant can be inappropriate at the next. Specifically, the number of possible solutions is infinite for

the IK of a limb, when the number of moving segments is larger than the Dof of the end-effectors. Similarly, an infinite number of solutions are possible for inverse dynamics, when the number of actuators (muscles) is larger than that of moving segments.

One of the early famous computational models of cerebellum is the cerebellar model articulation controller, which is based on the Marr–Albus ideas of the cerebellum (Albus 1975a,b; Pouget et al. 2000; Wolpert 1997). However, it was not originally proposed as a biological plausible approximation (Wolpert et al. 1998).

Given the unavailability of a training signal to the central nervous system (CNS), one of the most challenging tasks in modeling the cerebellum as an inverse model is acquiring an inverse dynamic model through motor learning. Kawato and colleagues (Kawato et al. 1987; Kawato and Gomi 1992) proposed a cerebellar feedback-error- learning model to solve this problem.

The learning process in the Kawato model is conducted in slow movement, after which the speed is increased . However, the model is insensitive to the noise because of the use of an open-loop control system, but one of the most important shortcomings is that it is not based on cerebellum physiology. Previous studies show that the cerebellum does not compute the inverse dynamic solutions and instead learns the forward one (Gentili et al. 2009).

The other inverse model for the cerebellum as a controller is based on the Smith predictor (Miall et al. 1993). This model is a forward model and has a delay structure that postpones the sensorimotor predictions for regulating the sensorimotor outcomes. In contrast to the Kawato model, the Smith model uses the inverse function, and the main (forward) function is trained. One of its advantages is that the error in this model is lower than that in the Kawato model (Wolpert et al. 1998).

The inverse problem can be bypassed by placing in a feedback loop a circuit that predicts the effect of motor orders, which provides it with a deterministic direct function (Barto et al. 1999; Darlot 1993; Houk 1996; Miall 1998; Miall et al. 1993; Miall and Wolpert 1996; Wolpert et al. 1998). The proposed theory holds that anticipating signal values is similar to the function of the cerebellar cortex and that computing approximate inverse functions is similar to the function of the entire cerebellum.

The model in Fig. 1 shows a feedback loop with a direct function $\Gamma(\alpha, u)$, a short delay of the feedback loop, and a delay to the motoneuron (α). The mathematical equations used to obtain $\Gamma(\alpha, u)$ with $x(t) = y(t)$ $x = y$ are as follows [i.e., $x(t)$ is the input signal and $y(t)$ is the output]:

$$y(t) = h(\alpha(t), u(t)) \tag{3}$$

$$\alpha(t+1) = x(t) + \Gamma(\alpha(t), u(t)) \tag{4}$$

This equation must be $\alpha(t+1) = \alpha(t)$ to obtain a fixed point.

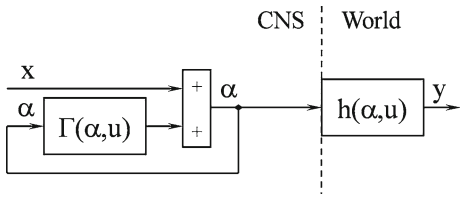

Fig. 1 Feedback model of a cerebellar cortex (Ebadzadeh and Darlot 2003; Ebadzadeh et al. 2005)

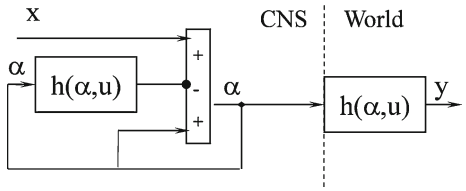

Fig. 2 Modified model of a cerebellar cortex (Ebadzadeh et al. 2005; Gentili et al. 2009)

Thus,

$$x(t) = \alpha(t) - \Gamma(\alpha(t), u(t)) \tag{5}$$

With $x = y$ and (3) and (5) combined,

$$x(t) = y(t) \Rightarrow \Gamma(\alpha(t), u(t)) = \alpha(t) - h(\alpha(t), u(t)) \tag{6}$$

Substituting (6) into (4) yields

$$\alpha(t+1) = g(\alpha(t)) = \alpha(t) + x(t) - h(\alpha(t), u(t)) \tag{7}$$

The modified model with respect to the equation for obtaining a fixed point is shown in Fig. 2.

The condition of the model stability (proven in "Appendix 1") is

$$0 < \frac{\partial h}{\partial \alpha} < 2 \tag{8}$$

The model is sometimes unstable because this condition is unsatisfied in every situation. The following modified model (Fig. 3), however, can satisfy the condition in every situation:

$$y(t) = h(\alpha(t), u(t)) \tag{9}$$

$$\alpha(t+1) = g(\alpha(t)) = \alpha(t) + B\,[x(t) - h(\alpha(t), u(t))] \tag{10}$$

The above equation must be $\alpha(t+1) = \alpha(t)$ to obtain a fixed point.

Thus,

$$B\,[x\,(t) - h\,(\alpha\,(t)\,, u\,(t))] = 0 \tag{11}$$

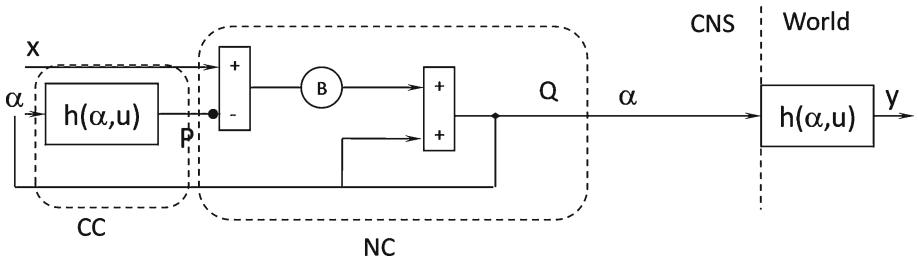

Fig. 3 Proposed model of a cerebellar cortex, where CC is the cerebellar cortex and NC is the cerebellar nucleus

and

$$x(t) = h(\alpha(t), u(t)) \tag{12}$$

Combining (9) and (12) yields $\Rightarrow x(t) = y(t)$.

Therefore, the proposed model works as an inverter. The stability condition of the proposed model (calculated in "Appendix 2") is shown in (22):

$$0 < B\frac{\partial g}{\partial \alpha} < 2 \tag{13}$$

Thus, the proposed model works in every situation because B can be varied as the stability condition is satisfied.

This circuit resembles the cerebellar pathways because positive and negative loops arranged in parallel evoke the excitatory mossy fibers that reach the cerebellar nuclei and cerebellar cortex. The inhibitory axons of the Purkinje cells project to the cerebellar nuclei. Thus, the predictor resembles the cerebellar cortex, and the summing element resembles the cerebellar nuclei (Figs. 3, 4). This anatomical interpretation matches the importance of the cerebellum in motor coordination and error compensation. The cerebellum is indispensable in achieving fast and precise coordinated movements and accurately perceiving body motion.

The input signal x reaches the summing element, which issues α through two pathways: a direct pathway that transmits this signal unchanged and an indirect pathway that processes the signal. This side circuit represents a motor part of the cerebellar pathways, which are situated outside the direct sensory or motor pathways.

The element, $h(\alpha, u)$, which is the predictor, represents the cerebellar cortex, which receives many sensory signals and efferent copies of motor signals through the mossy fibers. The cerebellar cortex processes these signals to increase the activity of Purkinje cells. This activity P is assumed to encode a dynamic signal that predicts the state of the body at the time the motor orders generate their effects (Figs. 3, 4).

The summing element placed immediately downstream $h(\alpha, u)$, where the signal of the positive loop is summed to the output P of $h(\alpha, u)$ and issues the signal Q, represents a group of neurons in a cerebellar nucleus (Figs. 3, 4). The same feedback signals reach the predictor and summing element, as the messages conveyed by the mossy fibers reach both the cerebellar cortex and cerebellar nuclei. The negative output of the predictor is comparable with the inhibitory projection of the Purkinje cells to the neurons of the cerebellar nuclei (Figs. 3, 4).

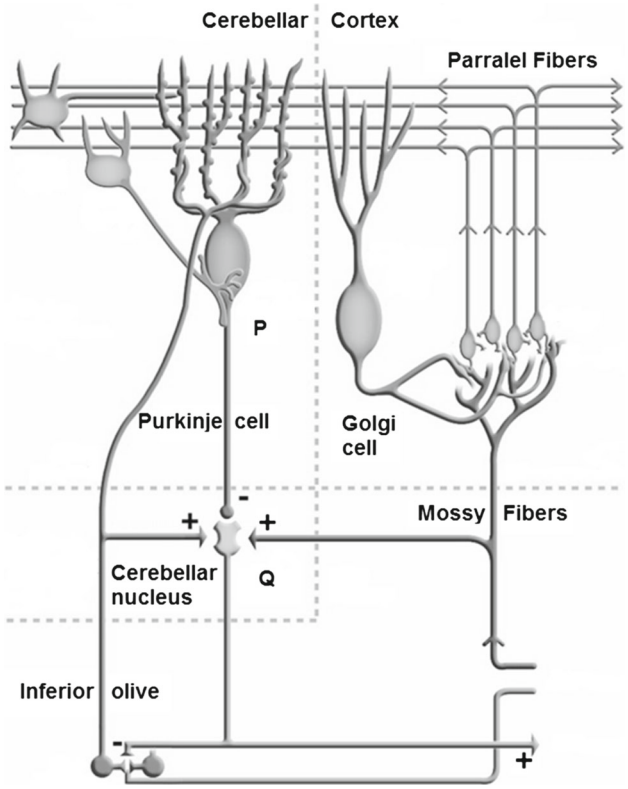

Fig. 4 Anatomy schematic diagram of human cerebellar cortex (Eccles et al. 1967; Kandel et al. 2000)

The cerebellum has three learning levels: (1) unsupervised learning in the glomerulus synapse, (2) supervised learning in the Purkinje synapse, and (3) supervised learning in the cerebellar nucleus (Jaberi et al. 2013; Jaeger 2013). The first and second learning methods have been previously reported (Bostan et al. 2013; Ebadzadeh et al. 2005; Schweighofer et al. 2013).

2.2 Fuzzy NN (FNN)

Calculating the optimal value of the model's parameter (B in Eq. 13) is computationally complex. ANN must be employed

to predict this value at any instant of time. FNN is used in this study to predict the optimal value of the model's parameter at any instant of time. The other ANNs that work as predictor cannot be employed because the nonlinear function (i.e., feedback loop in the model) learning methods can be implemented only in the FNNs.

The approximation of nonlinear functions can be modeled using a fuzzy rule based on a set of if–then rules defined as follows:

$$R^i = \text{if } x_1 \text{ is } A_1^i \text{ and } \ldots \text{ and } x_n \text{ is } A_n^i \text{ then } y \text{ is } B^i \tag{14}$$

where A_j^i and B^i are fuzzy sets, $x = (x_1, \ldots, x_n)^{\mathrm{T}}$ is the input variable, and y is the output variable of the fuzzy system.

Equation (15) is used to map a fuzzy set A' to a fuzzy set B' through the product inference engine.

$$\mu_{B'}(y) = \max_{i=1}^{m} \left[\sup \left(\mu_{A'}(x) \prod_{j=1}^{n} \mu_{A_j^i}(x_j)\, \mu_{B^i}(y) \right) \right]. \tag{15}$$

A real-value point x^* can be fuzzified by a singleton fuzzifier, which maps x^* to a fuzzy singleton A' with a membership value of 1 at x^* and 0 at other points:

$$\mu_{A'}(x) = \begin{cases} 1 & x = x^* \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

A defuzzifier is an algorithm that maps from a fuzzy set $B$to a crisp point y^*. y^{-i} is the center of the ith fuzzy set and w_i is its height. The center average defuzzifier determines y^* as

$$y^* = \frac{\sum_{i=1}^{m} y^{-i} w_i}{\sum_{i=1}^{m} w_i} \tag{17}$$

Fuzzy systems with the fuzzy rule base, a product inference engine, a singleton fuzzifier, and a center average defuzzifier have the following form for a fuzzy set B^i with a centery^{-i}:

$$f(x) = \frac{\sum_{i=1}^{m} y^{-i} \prod_{j=1}^{n} \mu_{A_j^i}(x_j)}{\sum_{i=1}^{m} \prod_{j=1}^{n} \mu_{A_j^i}(x_j)} \tag{18}$$

where x and $f(x)$ are the input and output of the fuzzy system (Wang 1999). The corresponding network of the fuzzy model can be built as shown in Fig. 5. The FNN has four layers. The output of each node in the first layer equals $\mu_{A_j^i}(x_j)$ and the membership value of fuzzy set A_j^i . The nodes in the second layer are used to calculate the product of the membership values and inputs in all the dimensions for each rule:

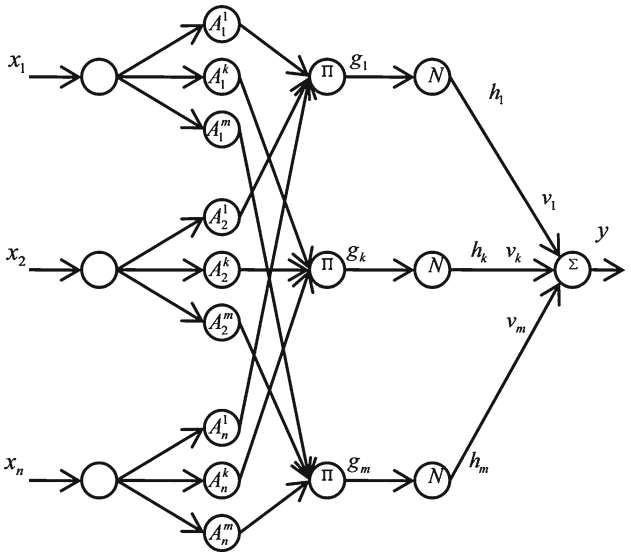

Fig. 5 Fuzzy neural network structure

$$g_i = \prod_{j=1}^{n} \mu_{A_j^i}(x_j). \tag{19}$$

The third layer is the normalization layer, where the output of node i is calculated as follows:

$$h_i = \frac{g_i}{\sum_{i=1}^{m} g_i} \tag{20}$$

Finally, the output of the ith node of the last (output) layer is the summation of its input values from the previous layer:

$$y = \sum_{i=1}^{M} v_i h_i, \tag{21}$$

where v_i s are the consequent parameters that should be learned through least-squares or gradient descent. For the Takagi–Sugeno–Kang fuzzy model, one layer before the output layer is added to replace the consequent parameters with a linear combination of inputs. Thus, the output of the network is calculated as follows:

$$y = \sum_{i=1}^{M} \left(c_0^i + c_1^i x_1 + \cdots + c_n^i x_n \right) h_i. \tag{22}$$

A hybrid learning algorithm determines the parameters in several neuro-fuzzy networks, wherein epochs involve forward and backward passes. All training data are presented to the network, and output weights are identified by the least-squares algorithm in the forward pass. Recursive least squares can also be used to determine the weights of the output layers (Kosko 1994; Malek et al. 2012).

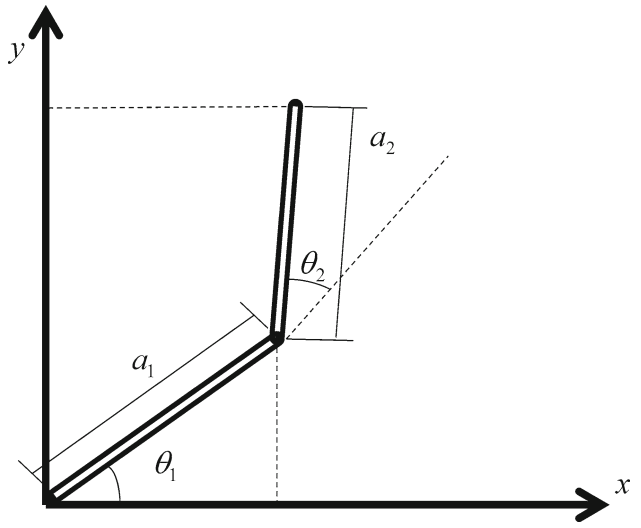

Fig. 6 Forward kinematics of the two-segmented arm

2.3 Proposed model to solve the IK problem

The control of the two-segmented arm is studied to test the proposed model. This examination requires the calculation of IK with two inputs and two outputs. The model in the previous section is proposed with one input and one output, so that the proposed model must be developed. In this section, the equation of the forward kinematics of the two segments is calculated, and the proposed model is developed.

With regard to Fig. 6, the equation of forward kinematics can be written as follows:

$$x = a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2) \tag{23}$$

$$y = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2) \tag{24}$$

where a_1 and a_2 are the lengths of the first and second segments, θ_1 and θ_2 are the angles of the same segments, and x and y are the Cartesian positions of the end-effector.

The velocity of x and y at any instant of time is calculated as follows:

$$\dot{x} = -a_1 \sin\theta_1 \dot{\theta}_1 - a_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \tag{25}$$

$$\dot{y} = a_1 \cos\theta_1 \dot{\theta}_1 + a_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \tag{26}$$

The developed model with regard to the previous equations for controlling a two-segmented arm is shown in Fig. 7.

$$\begin{aligned}\dot{\theta}_1(t+1) &= g_1(\dot{\theta}_1(t), \dot{\theta}_2(t)) \\ &= \dot{\theta}_1(t) + B_1(\dot{x}(t) - \dot{x}_{in}(t))\end{aligned} \tag{27}$$

$$\begin{aligned}\dot{\theta}_2(t+1) &= g_2(\dot{\theta}_1(t), \dot{\theta}_2(t)) \\ &= \dot{\theta}_2(t) + B_2(\dot{y}(t) - \dot{y}_{in}(t))\end{aligned} \tag{28}$$

The equations must be $\dot{\theta}_1(t+1) = \dot{\theta}_1(t)$ and $\dot{\theta}_2(t+1) = \dot{\theta}_2(t)$ to obtain a fixed point.

Thus,

$$B_1(\dot{x}(t) - \dot{x}_{in}(t)) = 0 \quad \Rightarrow \quad \dot{x}(t) = \dot{x}_{in}(t) \tag{29}$$

$$B_2(\dot{y}(t) - \dot{y}_{in}(t)) = 0 \quad \Rightarrow \quad \dot{y}(t) = \dot{y}_{in}(t) \tag{30}$$

The results verify that the developed model acts as an inverter.

The stability condition of the developed model (calculated in the "Appendix 3") is shown in Eq. (31):

$$-4 < \left(B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2}\right) \pm \sqrt{\left(B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2}\right)^2 + 4B_1 B_2 \frac{\partial \dot{x}}{\partial \dot{\theta}_2} \frac{\partial \dot{y}}{\partial \dot{\theta}_1}} < 0 \tag{31}$$

The coefficients B_1 and B_2 must satisfy the condition of Eq. (31) to prove the stability of the proposed model. The values of these coefficients depend on Eqs. (25) and (26). The appropriate values of B_1 and B_2 depend on angles θ_1 and θ_2 and their velocity at any instant of time, which indicates

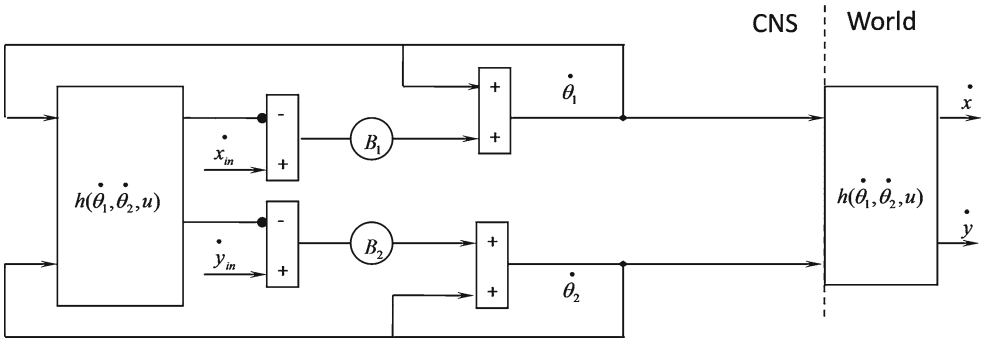

Fig. 7 Developed cerebellar cortex model for solving two-segmented arm IK

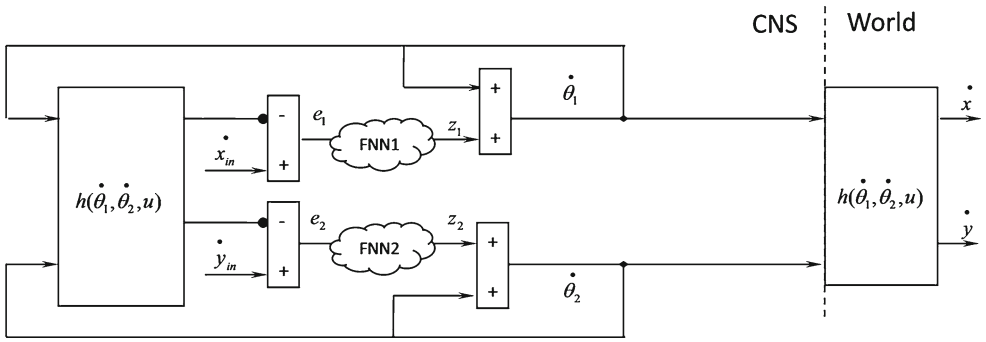

Fig. 8 Using FNN in the developed cerebellar cortex model

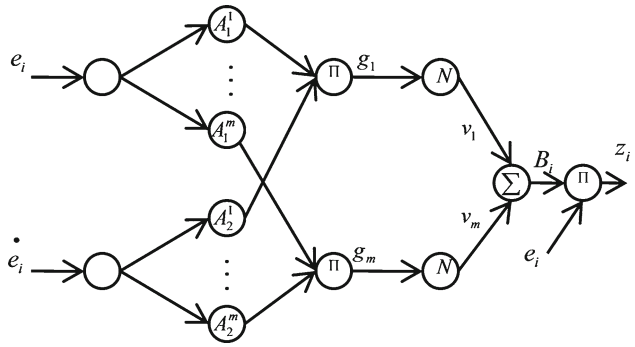

Fig. 9 The structure of FNN used in the model

that the optimal values of B_1 and B_2 are dynamic and different at any instant of time. This computational complexity necessitates the calculation of optimal values with an FNN.

Figure 8 shows the location of two FNNs, and Fig. 9 shows the proposed FNN for learning B_1 and B_2.

The function $h = (\dot{\theta}_1, \dot{\theta}_2, u)$ is unknown, so that the weights v_i of the network must be learned. One of the most important reasons for selecting FNNs is the learning method, which runs only through these networks. The FNN used in this study is a one-layer network with a normal triangular fuzzy set. Normal triangular fuzzy sets are created as shown in Fig. 10, where μ and v are the membership function and weight, respectively.

Considering $y = f(x)$, the output of the network is calculated according to $y = \sum v_i \mu_{A_i}(x)$. If the condition $v_i = f(m_i)$ is true, then $f(x)$ can be defined as a piecewise linear function. Given that $f(x)$ is an unknown function, the weight v_i should be learned by the network. Gradient and pseudo-inverse learning methods are inapplicable because the functions in the proposed model are nonlinear and cannot be differentiated as a result of the feedback loop.

The FNN has several features that make it an appropriate object of learning through the proposed learning method:

1. The point values in the fuzzy sets equal the function ($\mu = 1$).

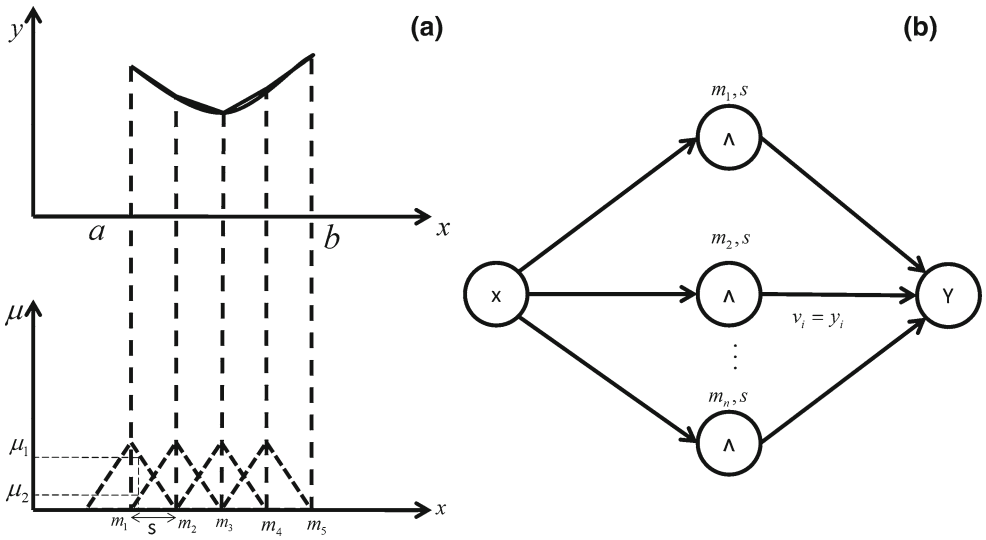

Fig. 10 **a** Normal triangular fuzzy sets and **b** rules for learning the weights of the network

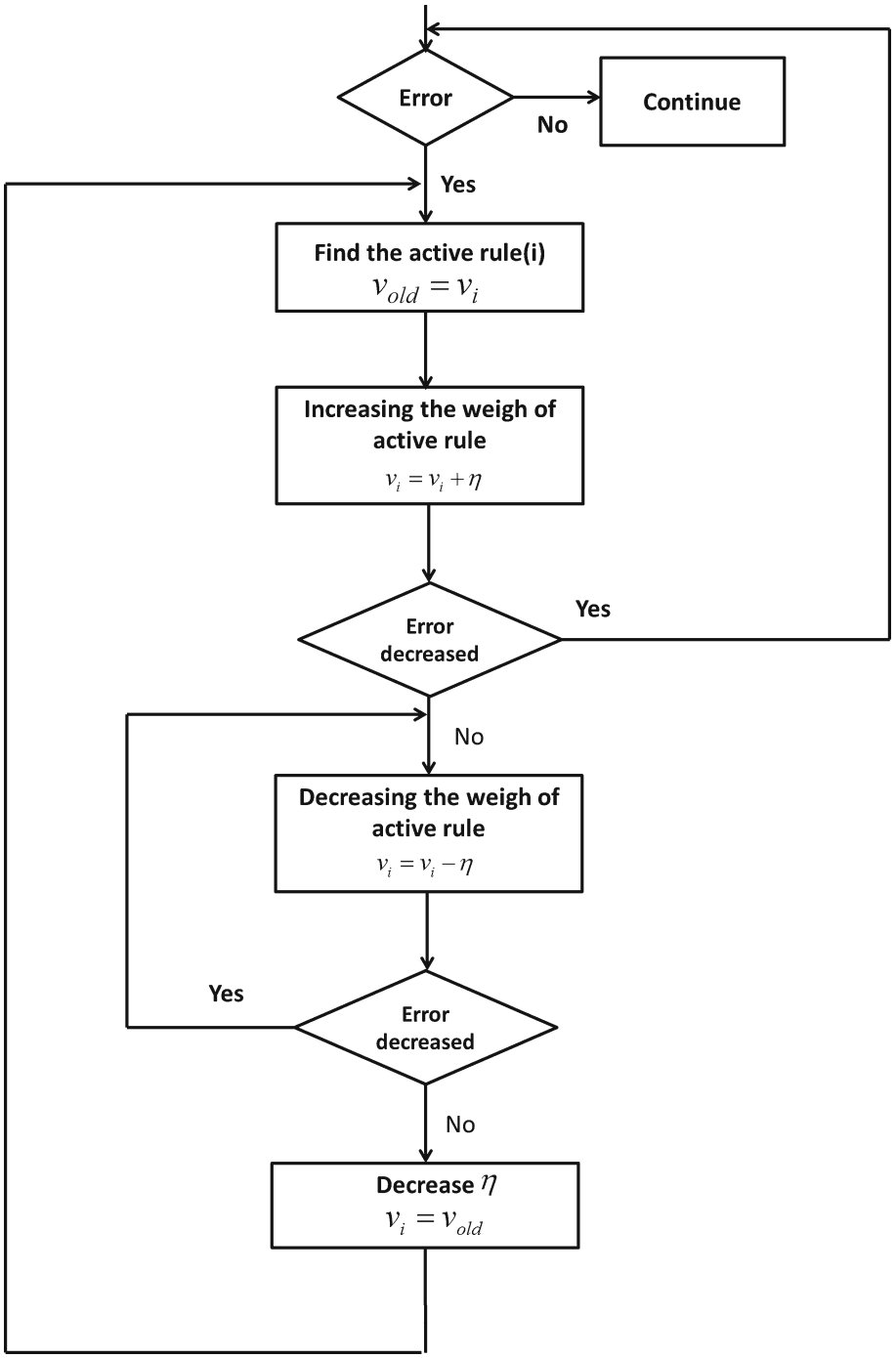

Fig. 11 Flowchart of learning algorithm used for learning the weights of FNN

2. The points in the s intervals are a linear combination of the preceding and following output values ($y = \mu_1 y_1 + \mu_2 y_2$).
3. The fuzzy sets in this study are normal; hence, $\mu_1 + \mu_2 = 1$.

Therefore, only a maximum of two rules is equal to one, and the other rules are equal to zero at every instant of time.

We use FNN in this study because the nodes in the other networks are active at each step, which entails high cost. Our solution to this problem is shown as a learning algorithm in Fig. 11.

As previously mentioned, we cannot use differentiation because of the feedback loop. Therefore, whether or not the weight of the active rule at each step must be increased or decreased cannot be determined. A solution to this problem is

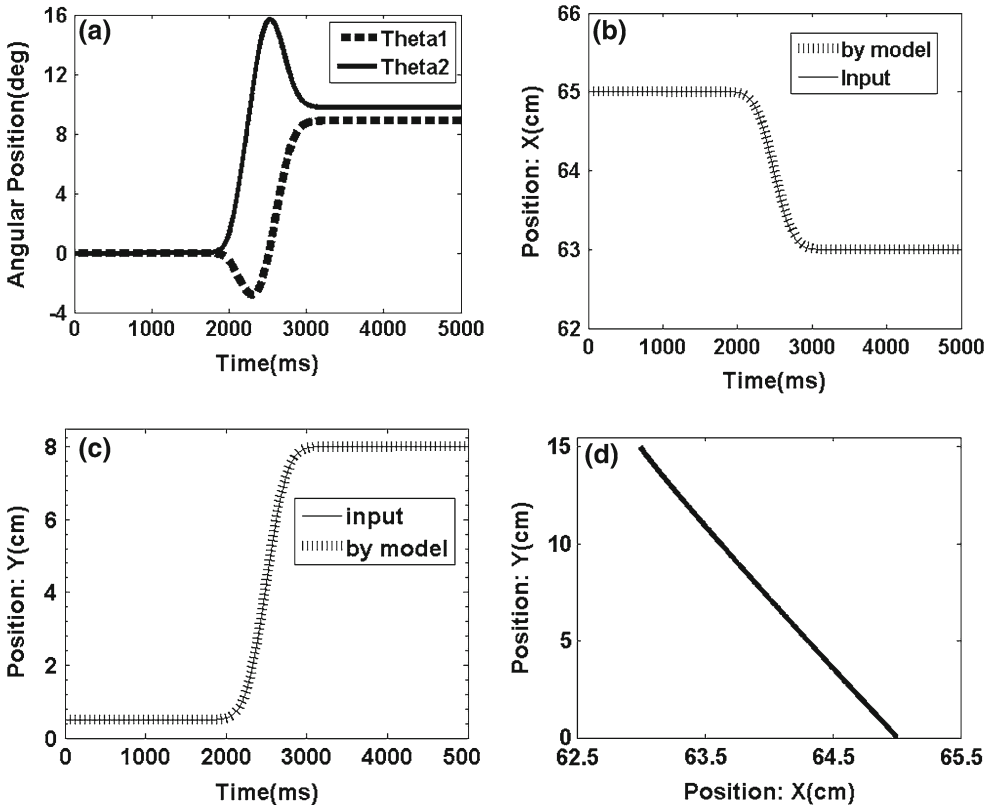

Fig. 12 Movement profile of the two-segmented arm from point (65,0) to point (63,15). **a** Angles of the two-segmented arm that reach the desire point. **b** Arm displacement in the x axis direction from point 65 to point 63. **c** Arm displacement in the y axis direction from point 0 to point 15. **d** Trajectory in the joint coordinates

to increase the weight of the active rule and subsequently calculate the error (Fig. 11). If the error decreases, the increasing step is continued until the error becomes constant or drops. If the error increases, the weight must be decreased until the error is stabilized.

3 Results and discussion

Three experiments are designed and implemented in MATLAB (R2008a) 7.6.0.324 to evaluate the proposed model. The lengths of the first and second segments are considered to be $a_1 = 0.35$ m and $a_2 = 0.3$ m, respectively. The Gaussian function considers the velocity of θ to be the input of the model, and the velocity of movement is considered to be $\sigma = 0.2$.

The end-effector of the two-segmented robot moves from point (65, 0) to point (63, 15) in the first experiment (Fig. 12). The movement exhibits a trajectory in the joint coordinates, which is expected to be a straight line in the best-case scenario. Figures 12a, 13 and 14a show the angles of each arm segment that reach the desired position.

The end-effector of the two-segmented robot moves from point (65,0) to point (60,20) and from point (65,0) to point (50,40) in the second and third experiments, respectively (Figs. 13, 14).

The models provide satisfying results, and the trajectory in the joint coordinates in all three experiments is approximately a straight line (Figs. 13, 14). These results show that the error of the model is acceptable in all cases (i.e., the mean square error is below the 0.0001).

Thus, the results of experiments show that the proposed approach is a feasible option for the real-time path planning and precise control of robots.

4 Conclusion

The IK problem is solved through a method that resembles cerebellar anatomy and function. Previous cerebellum-inspired solutions were based on a direct function and a recursive model that remains steady only under certain conditions. Therefore, they cannot be applied to robot movement with high Dof.

We used mathematical analysis to propose a modified model that is stable under all conditions because only one parameter is determined. This dynamic parameter varies at any instant. Therefore, we use an FNN with a particular learning method.

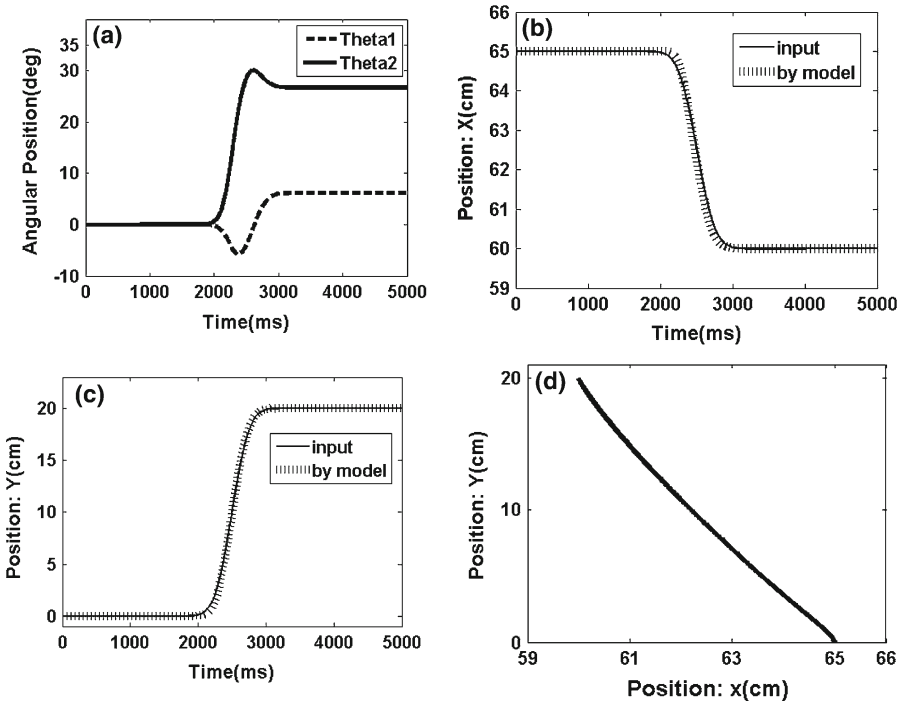

Fig. 13 Movement profile of the two-segmented arm from point (65,0) to point (60,20). **a** Angles of the two-segmented arm that reached the desired point. **b** Arm displacement in the x axis direction from point 65 to point 60. **c** Arm displacement in the y axis direction from point 0 to point 20. **d** Trajectory in the joint coordinates

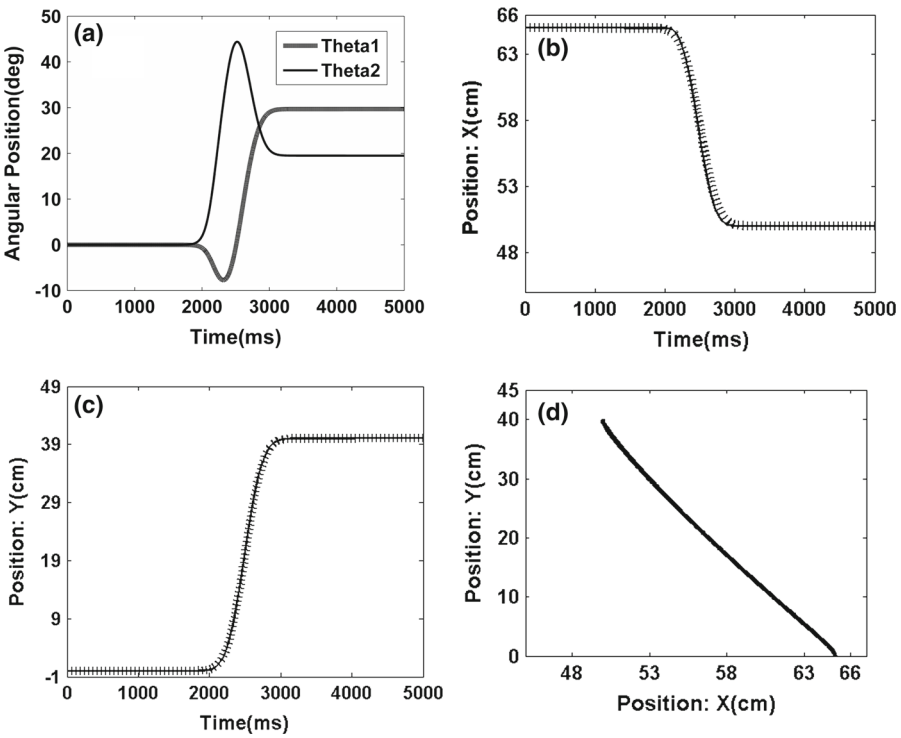

Fig. 14 Movement profile of the two-segmented arm from point (65,0) to point (50,40). **a** Angles of the of the two-segmented arm that reach the desired point. **b** Arm displacement in the x axis direction from point 65 to point 50. **c** Arm displacement in the y axis direction from point 0 to point 40. **d** Trajectory in the joint coordinates

The proposed model is developed and modified for a simple two-segmented arm to show its feasibility. Two model parameters are approximated by FNN because of its specific learning method. The model has an acceptable and reliable performance in solving the IK problem of the two-segmented arm.

Moreover, the proposed model can be generalized for all functions because it is independent of the calculations for the inverse function, which is proven mathematically. This approach can be used for the prediction of IK solutions for any kind of robot regardless of the geometry and Dof associated with it.

Acknowledgments This study was supported by High Impact Research UM/MOHE/HIR Project No. D000010-16001 from the University of Malaya.

Appendix 1

There are some methods obtaining the model stability. One of them is to linearize it in its fixed point.

The linear equation of $g(\alpha(t))$ in α_0 is as follows:

$$g(\alpha(t)) - g(\alpha_0) = \left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0} (\alpha(t) - \alpha_0) \tag{32}$$

With consideration of Eq. (16),

$$g(\alpha(t)) = \alpha(t+1) \tag{33}$$

And because α_0 is fixed point of $g(\alpha(t))$,

$$g(\alpha_0) = \alpha_0 \tag{34}$$

With (32), (33), and (34) combined,

$$\alpha(t+1) - \alpha_0 = \left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0} (\alpha(t) - \alpha_0) \tag{35}$$

And with replacement $t+1$ in (35),

$$\alpha(t+2) - \alpha_0 = \left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0} (\alpha(t+1) - \alpha_0) \tag{36}$$

With (35) and (36) combined,

$$\alpha(t+2) - \alpha_0 = \left(\left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0}\right)^2 (\alpha(t) - \alpha_0) \tag{37}$$

Thus,

$$\alpha(t+n) - \alpha_0 = \left(\left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0}\right)^n (\alpha(t) - \alpha_0) \tag{38}$$

For stability must be:

$$\alpha(t+n) = \alpha_0 \tag{39}$$

This means that

$$-1 < \left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0} < 1 \tag{40}$$

With derivation of Eq. (16),

$$\frac{\partial g}{\partial \alpha} = 1 - \frac{\partial h}{\partial \alpha} \tag{41}$$

With (40) and (41) combined,

$$0 < \frac{\partial h}{\partial \alpha} < 2 \tag{42}$$

Appendix 2

The linear equation of $g(\alpha(t))$ in α_0 is as follows:

$$g(\alpha(t)) - g(\alpha_0) = \left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0} (\alpha(t) - \alpha_0) \tag{43}$$

With consideration of Eq. (19),

$$g(\alpha(t)) = \alpha(t+1) \tag{44}$$

And because α_0 is fixed point of $g(\alpha(t))$,

$$g(\alpha_0) = \alpha_0 \tag{45}$$

With (43), (44) and (45) combined,

$$\alpha(t+1) - \alpha_0 = \left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0} (\alpha(t) - \alpha_0) \tag{46}$$

And with replacement $t+1$ in (46),

$$\alpha(t+2) - \alpha_0 = \left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0} (\alpha(t+1) - \alpha_0) \tag{47}$$

With (46) and (47) combined,

$$\alpha(t+2) - \alpha_0 = \left(\left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0}\right)^2 (\alpha(t) - \alpha_0) \tag{48}$$

Thus,

$$\alpha(t+n) - \alpha_0 = \left(\left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0}\right)^n (\alpha(t) - \alpha_0) \tag{49}$$

For stability must be :

$$\alpha(t+n) = \alpha_0 \tag{50}$$

This means that

$$-1 < \left.\frac{\partial g}{\partial \alpha}\right|_{\alpha_0} < 1 \tag{51}$$

With derivation of Eq. (19),

$$\frac{\partial g}{\partial \alpha} = 1 - B\frac{\partial h}{\partial \alpha} \tag{52}$$

With (51) and (52) combined,

$$0 < B\frac{\partial h}{\partial \alpha} < 2$$

Appendix 3

The linear equation of $g_1(\dot{\theta}_1(t), \dot{\theta}_2(t))$ and $g_2(\dot{\theta}_1(t), \dot{\theta}_2(t))$ in $\dot{\theta}_{10}$ and $\dot{\theta}_{20}$,

$$\begin{aligned} &g_1(\dot{\theta}_1(t), \dot{\theta}_2(t)) - g_1(\dot{\theta}_{10}, \dot{\theta}_{20}) \\ &= \left.\frac{\partial g_1}{\partial \dot{\theta}_1}\right| (\dot{\theta}_1(t) - \dot{\theta}_{10}) + \left.\frac{\partial g_1}{\partial \dot{\theta}_2}\right| (\dot{\theta}_2(t) - \dot{\theta}_{20}) \end{aligned} \tag{53}$$

$$\begin{aligned} &g_2(\dot{\theta}_1(t), \dot{\theta}_2(t)) - g_2(\dot{\theta}_{10}, \dot{\theta}_{20}) \\ &= \left.\frac{\partial g_2}{\partial \dot{\theta}_1}\right| (\dot{\theta}_1(t) - \dot{\theta}_{10}) + \left.\frac{\partial g_2}{\partial \dot{\theta}_2}\right| (\dot{\theta}_2(t) - \dot{\theta}_{20}) \end{aligned} \tag{54}$$

With consideration of Eqs. (27), (28),

$$g_1(\dot{\theta}_1(t), \dot{\theta}_2(t)) = \dot{\theta}_1(t+1) \tag{55}$$

$$g_2(\dot{\theta}_1(t), \dot{\theta}_2(t)) = \dot{\theta}_2(t+1) \tag{56}$$

Because $\dot{\theta}_{1_0}$ and $\dot{\theta}_{2_0}$ are fixed points of $g_1(\dot{\theta}_1(t), \dot{\theta}_2(t))$and $g_2(\dot{\theta}_1(t), \dot{\theta}_2(t))$, respectively,

$$g_1(\dot{\theta}_{10}, \dot{\theta}_{20}) = \dot{\theta}_{10} \tag{57}$$

$$g_2(\dot{\theta}_{10}, \dot{\theta}_{20}) = \dot{\theta}_{20} \tag{58}$$

With (53) to (58) combined,

$$\begin{aligned} \dot{\theta}_1(t+1) - \dot{\theta}_{1_0} = \left.\frac{\partial g_1}{\partial \dot{\theta}_1}\right| \left(\dot{\theta}_1(t) - \theta_{1_0} + \left.\frac{\partial g_1}{\partial \dot{\theta}_2}\right)\right| \\ \times \left(\dot{\theta}_2(t) - \dot{\theta}_{2_0}\right) \end{aligned} \tag{59}$$

$$\begin{aligned} \dot{\theta}_2(t+1) - \dot{\theta}_{2_0} = \left.\frac{\partial g_2}{\partial \dot{\theta}_1}\right| \left(\dot{\theta}_1(t) - \theta_{1_0} + \left.\frac{\partial g_2}{\partial \dot{\theta}_2}\right)\right| \\ \times \left(\dot{\theta}_2(t) - \dot{\theta}_{2_0}\right) \end{aligned} \tag{60}$$

$$\begin{bmatrix} \dot{\theta}_1(t+1) - \theta_{1_0} \\ \dot{\theta}_2(t+1) - \theta_{2_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial \dot{\theta}_1} & \frac{\partial g_1}{\partial \dot{\theta}_2} \\ \frac{\partial g_2}{\partial \dot{\theta}_1} & \frac{\partial g_2}{\partial \dot{\theta}_2} \end{bmatrix}^n \begin{bmatrix} \dot{\theta}_1(t) - \dot{\theta}_{1_0} \\ \dot{\theta}_2(t) - \dot{\theta}_{2_0} \end{bmatrix} \tag{61}$$

And at $(t+n)$,

$$\begin{bmatrix} \dot{\theta}_1(t+n) - \theta_{1_0} \\ \dot{\theta}_2(t+n) - \theta_{2_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial \dot{\theta}_1} & \frac{\partial g_1}{\partial \dot{\theta}_2} \\ \frac{\partial g_2}{\partial \dot{\theta}_1} & \frac{\partial g_2}{\partial \dot{\theta}_2} \end{bmatrix}^n \begin{bmatrix} \dot{\theta}_1(t) - \dot{\theta}_{1_0} \\ \dot{\theta}_2(t) - \dot{\theta}_{2_0} \end{bmatrix} \tag{62}$$

With considering (62),

$$\begin{bmatrix} \frac{\partial g_1}{\partial \dot{\theta}_1} & \frac{\partial g_1}{\partial \dot{\theta}_2} \\ \frac{\partial g_2}{\partial \dot{\theta}_1} & \frac{\partial g_2}{\partial \dot{\theta}_2} \end{bmatrix} = A \tag{63}$$

As known, 'A' is a semi-positive matrix, so

$$A = \phi^T \Lambda \phi \;\Rightarrow\; A^n = \phi^T \Lambda^n \phi \tag{64}$$

Where ϕ Eigen is vector matrix and Λ is eigenvalue matrix:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \Rightarrow \Lambda^n = \begin{bmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{bmatrix} \tag{65}$$

For stability must be:

$$\begin{bmatrix} \dot{\theta}_1(t+n) - \theta_{1_0} \\ \dot{\theta}_2(t+n) - \theta_{2_0} \end{bmatrix} = 0 \Rightarrow \Lambda^n = 0 \Rightarrow -1 < \lambda_i < 1 \tag{66}$$

Considering Eq. 27 and 28,

$$\frac{\partial g_1}{\partial \dot{\theta}_1} = 1 + B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} \tag{67}$$

$$\frac{\partial g_1}{\partial \dot{\theta}_2} = B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_2} \tag{68}$$

$$\frac{\partial g_2}{\partial \dot{\theta}_1} = B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_1} \tag{69}$$

$$\frac{\partial g_2}{\partial \dot{\theta}_2} = 1 + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2} \tag{70}$$

With combining above equations,

$$A = \begin{bmatrix} 1 + B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} & B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_2} \\ B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_1} & 1 + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2} \end{bmatrix} \tag{71}$$

To calculate eigenvalue of matrix 'A,'

$$\left(1 - \lambda + B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1}\right)\left(1 - \lambda + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2}\right) - B_1 B_2 \frac{\partial \dot{x}}{\partial \dot{\theta}_2} \frac{\partial \dot{y}}{\partial \dot{\theta}_1} = 0 \tag{72}$$

$$(1-\lambda)^2 + \left(B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2}\right)(1-\lambda) - B_1 B_2 \frac{\partial \dot{x}}{\partial \dot{\theta}_2} \frac{\partial \dot{y}}{\partial \dot{\theta}_1} = 0 \tag{73}$$

$$1 - \lambda = \frac{-\left(B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2}\right) \pm \sqrt{\left(B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2}\right)^2 + 4 B_1 B_2 \frac{\partial \dot{x}}{\partial \dot{\theta}_2} \frac{\partial \dot{y}}{\partial \dot{\theta}_1}}}{2} \tag{74}$$

$$\lambda = 1 + \frac{\left(B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2}\right)}{2} \pm \frac{1}{2} \sqrt{\left(B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2}\right)^2 + 4 B_1 B_2 \frac{\partial \dot{x}}{\partial \dot{\theta}_2} \frac{\partial \dot{y}}{\partial \dot{\theta}_1}} \tag{75}$$

With considering the above results and Eq. (66),

$$-2 < \lambda_i < 0 \tag{76}$$

$$-4 < \left(B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2}\right) \pm \sqrt{\left(B_1 \frac{\partial \dot{x}}{\partial \dot{\theta}_1} + B_2 \frac{\partial \dot{y}}{\partial \dot{\theta}_2}\right)^2 + 4 B_1 B_2 \frac{\partial \dot{x}}{\partial \dot{\theta}_2} \frac{\partial \dot{y}}{\partial \dot{\theta}_1}} < 0 \tag{77}$$

References

Alavandar S, Nigam M (2008) Neuro-fuzzy based approach for inverse kinematics solution of industrial robot manipulators. Int J Comput Commun Control 3:224–234

Albus JS (1975a) Data storage in the cerebellar model articulation controller (CMAC). J Dyn Syst Meas Control 97:228–233

Albus JS (1975b) A new approach to manipulator control: the cerebellar model articulation controller (CMAC). J Dyn Syst Meas Control 97:220–227

Ali MA, Park HA, Lee CG (2010) Closed-form inverse kinematic joint solution for humanoid robots. In: 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 704–709

Arefi R, Sadigh M (2011) Fuzzy inverse kinematics algorithm for man and machine cooperation. In: 2011 IEEE international conference on mechatronics (ICM). IEEE, pp. 398–402

Barto AG, Fagg AH, Sitkoff N, Houk JC (1999) A cerebellar model of timing and prediction in the control of reaching. Neural Comput 11:565–594

Bostan AC, Dum RP, Strick PL (2013) Cerebellar networks with the cerebral cortex and basal ganglia. Trends Cogn Sci 17(5):241–254. doi:10.1016/j.tics.2013.03.003

Cannon S, Robinson D (1987) Loss of the neural integrator of the oculomotor system from brain stem lesions in monkey. J Neurophysiol 57:1383–1409

Chiddarwar SS, Ramesh Babu N (2010) Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach. Eng Appl Artif Intell 23:1083–1092

Darlot C (1993) The cerebellum as a predictor of neural messages–I. The stable estimator hypothesis. Neuroscience 56:617–646

Dash K, Choudhury B, Khuntia A, Biswal B (2011) A neural network based inverse kinematic problem. In: 2011 IEEE on recent advances in intelligent computational systems (RAICS). IEEE, pp 471–476

de Jesús Rubio J, Aquino V, Figueroa M (2013) Inverse kinematics of a mobile robot. Neural Comput Appl 23(1):187–194. doi:10.1007/s00521-012-0854-0

Ebadzadeh M, Darlot C (2003) Cerebellar learning of bio-mechanical functions of extra-ocular muscles: modeling by artificial neural networks. Neuroscience 122:941–966

Ebadzadeh M, Tondu B, Darlot C (2005) Computation of inverse functions in a model of cerebellar and reflex pathways allows to control a mobile mechanical segment. Neuroscience 133:29–49

Eccles JC, Ito M, Szentágothai J (1967) The cerebellum as a neuronal machine. Springer, New York

Gentili RJ, Papaxanthis C, Ebadzadeh M, Eskiizmirliler S, Ouanezar S, Darlot C (2009) Integration of gravitational torques in cerebellar pathways allows for the dynamic inverse computation of vertical pointing movements of a robot arm. PloS ONE 4:e5176

Hasan AT, Hamouda AMS, Ismail N, Al-Assadi H (2006) An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 DOF serial robot manipulator. Adv Eng Softw 37:432–438

Hasan AT, Ismail N, Hamouda AMS, Aris I, Marhaban MH, Al-Assadi H (2010) Artificial neural network-based kinematics Jacobian solution for serial manipulator passing through singular configurations. Adv Eng Softw 41:359–367

Houk AGBJC (1996) A predictive switching model of cerebellar movement control. In: Proceedings of the 1995 conference. Kaufmann, p 138

Jaberi J, Gambrell K, Tiwana P, Madden C, Finn R (2013) Long-term clinical outcome analysis of poly-methyl-methacrylate cranioplasty for large skull defects. J Oral Maxillofac Surg 71:e81–e88

Jaeger D (2013) Cerebellar nuclei and cerebellar learning. Handbook of the cerebellum and cerebellar disorders. Springer, Berlin

Kandel ER, Schwartz JH, Jessell TM (2000) Principles of neural science. McGraw-Hill, New York

Kanoun O, Laumond J-P, Yoshida E (2011) Planning foot placements for a humanoid robot: a problem of inverse kinematics. Int J Robot Res 30:476–485

Karlik B, Aydin S (2000) An improved approach to the solution of inverse kinematics problems for robot manipulators. Eng Appl Artif Intell 13:159–164

Kawato M, Furukawa K, Suzuki R (1987) A hierarchical neural-network model for control and learning of voluntary movement. Biolog Cybern 57:169–185

Kawato M, Gomi H (1992) The cerebellum and VOR/OKR learning models. Trends Neurosci 15:445–453

Köker R (2013) A genetic algorithm approach to a neural-network based inverse kinematics solution of robotic manipulators based on error minimization. Inf Sci 222:528–543. doi:10.1016/j.ins.2012.07.051

Kosko B (1994) Fuzzy systems as universal approximators. IEEE Trans Comput 43:1329–1333

Kumar S, Behera L, McGinnity TM (2010) Kinematic control of a redundant manipulator using an inverse-forward adaptive scheme with a KSOM based hint generator. Robot Auton Syst 58:622–633

Malek H, Ebadzadeh MM, Rahmati M (2012) Three new fuzzy neural networks learning algorithms based on clustering, training error and genetic algorithm. Appl Intell 37:280–289

Miall R (1998) The cerebellum, predictive control and motor coordination. Sens Guid Mov 218:272–290

Miall R, Weir D, Wolpert D, Stein J (1993) Is the cerebellum a Smith predictor? J Mot Behav 25:203–216

Miall R, Wolpert DM (1996) Forward models for physiological motor control. Neural Netw 9:1265–1279

Oyama E, Agah A, MacDorman KF, Maeda T, Tachi S (2001) A modulator neural network architecture for inverse kinematics model learning. Neurocomputing 38–40:797–805

Pouget A, Dayan P, Zemel R (2000) Information processing with population codes. Nat Rev Neurosci 1:125–132

Reinhart RF, Steil JJ (2011) Neural learning and dynamical selection of redundant solutions for inverse kinematic control. In: 2011 11th IEEE-RAS international conference on humanoid robots (Humanoids). IEEE, pp 564–569

Schweighofer N, Lang EJ, Kawato M (2013) Role of the olivo-cerebellar complex in motor learning and control. Frontiers Neural Circuits 7:94. doi:10.3389/fncir.2013.00094

Tikhonov A, Arsenin VY (1979) Methods for solving ill-posed problems. Nauka, Moscow

Wang L-X (1999) A course in fuzzy systems. Prentice-Hall Press, Englewood Cliffs NJ

Wang X, Wang L, Pan C, Zhang Y, Tang W, Zhang X (2010) Inverse kinematics analysis of multi-legged walking robots based on hand-foot-integration mechanism. In: 2010 international conference on mechatronics and automation (ICMA). IEEE, pp 1184–1189. doi:10.1109/ICMA.2010.5587954

Wolpert DM (1997) Computational approaches to motor control. Trends Cogn Sci 1:209–216

Wolpert DM, Miall RC, Kawato M (1998) Internal models in the cerebellum. Trends Cogn Sci 2:338–347

Wu W, Rao S (2007) Uncertainty analysis and allocation of joint tolerances in robot manipulators based on interval analysis. Reliab Eng Syst Saf 92:54–64

Xia Y, Wang J (2001) A dual neural network for kinematic control of redundant robot manipulators. IEEE Trans Syst Man Cybern Part B Cybern 31:147–154

Zhang H, Paul RP (1991) A parallel inverse kinematics solution for robot manipulators based on multiprocessing and linear extrapolation. IEEE Trans Robot Autom 7:660–669